SCHOOL OF MATHEMATICAL AND COMPUTER SCIENCES

Computer Science

F28PL

Programming Languages

Semester 1 201516

Duration: Two Hours

ANSWER THREE QUESTIONS

1. (a) Clearly write the ML types of the following expressions, or if the expression has no ML type, explain why: (5)

1. `0`
2. `0.0`
3. `"zero"`
4. `fn x => 1/x`
5. `fn f => fn x => x`

(b) Explain any similarities, and differences, between the following ML types:

- `('a * 'b) -> 'c`
- `'a -> 'b -> 'c`
- `('a -> 'b) -> 'c` (3)

(c) 1. Convert the following mathematical specification of a *factorial* function on real numbers directly into ML code for a function `fact`: (3)

$$0.0! = 1.0 \qquad n! = n * (n{-}1.0)! \quad (\text{if } n \neq 0.0)$$

2. Write the ML type of `fact`. (2)
3. State whether `fact` is tail recursive, and if so, why. (2)
(Answers that just say 'Yes' or 'No', without demonstrating understanding of why, may score zero marks.)
4. What results when we compute `fact 1.1`, and why? (2)

(d) Write an ML function to implement the following summation—up to some integer n—which approximates $1/\pi$ with very high efficiency (this is the basis of modern approximations of π): (3)

$$\frac{1}{\pi} = \frac{2 * \sqrt{2}}{9801} \sum_{k \geq 0} \frac{(4 * k)!(1103 + 26390 * k)}{(k!)^4 * 396^{4*k}}.$$

(We do not bother to indicate real numbers with '.0' above, because this is maths, not programming.)

Note that if f is a function from integers to reals, then

$$\sum_{k \geq 0} f(k) \quad \text{means} \quad f(0) + f(1) + f(2) + \ldots.$$

You may find it useful to define a helper function

```
sumto : (int -> real) -> int -> real
```

You may assume functions for multiplication, division, square root `sqrt`, factorial `fact`, and power `pow`. Minor type errors between `int` and `real` will not be marked down *unless* they indicate some lack of understanding. Answers that are illegible or not clearly laid out, however, may be marked down if the examiner cannot easily read them.

2. (a) State the output of the following programs and explain why, or, if the program terminates with an error, state what that error is and why it arises:

1. `range(0,10,2)[1]` (2)
2. `{ 1:2 , 3:4 }[3]` (2)
3. `{ [1]:2, [3]:4 }[[3]]` (2)
4.
```
{2: ["tom", "and", "jerry"],
 0: ["bugs", "bunny"],
 1: ["micky", "and", "minnie"]}[1][-1][1:]
```
(2)

(b) Consider the following simplified programming model of married life:

```
1  husband_todo = [ ]
2  tasks = [ "rubbish", "dishwasher", "laundry",
3            "vacuum", "nappy", 'feed', "clean" ]
4
5  defn wife()
6     husband_todo.append(tasks[len(husband_todo) % 7])
7     if len(husband_todo)>7 then:
8     husband_todo.append("apologise")
9
10 while true:
11    wife()
12 #  husband()
13    print(husband_todo,"\n")
```

1. This program is defective and contains five errors. State on what line number(s) these errors occur, what they are, and how to correct them. (5)
2. Describe in specific detail the behaviour and expected output of the program above. (3)
3. Design and write out a syntactically correct Python function `husband()` that would ensure, if the call to `husband` on line 12 is commented out, that `"apologise"` is never added to `husband_todo`. (1)

(c) State the output of the following program, the meaning of that output, and how the program works:

```
(lambda r: [x for x in r if x not in
             [x*y for x in r for y in r]])(range(2,15))
```

(3)

3. (a) Compare and contrast the typing systems of ML, Python, and Prolog. Your answer should clearly indicate of each language:

- what the type syntaxes look like for each language (give examples),
- when type errors are raised (compile-time, run-time, both, neither, or something else),
- the relative sophistication and complexity of the type systems including precise details of the structure of the types, and
- to what degree they are polymorphic or not.

Answers that convince the examiner that you have a specific, practical, and full understanding of the type systems and how they work in all three languages, will score full marks. So be specific, full, and illustrate your answer with concrete type syntax where appropriate. (8)

(b) "All the languages on this course are bunk. Just use Java."

Discuss the claim above—(*bunk* means *unnecessary and pointless*)—giving two detailed arguments for, and two detailed arguments against. Your reasons need not be confined to specific programming issues and could include social, financial, and other reasons. (4)

(c) Explain the meaning of the term *global state*. (2)

(d) ML, Python, and Prolog all have forms of global state. Describe them, being (as always) specific, detailed, and giving concrete syntax examples where appropriate. (2)

(e) Explain the terms *declarative programming paradigm* and *imperative programming paradigm*. Name one specific advantage and one specific disadvantage of each paradigm. (4)

4. Below, when we write 'State the output of', you should include output obtained by tabbing through possible multiple instantiations suggested by the interpreter (thus, questions that only state the first instantiation, without the others, may lose marks). If the program loops, crashes or in some other way does not work, your answer must be specific about the nature of the loop and (in overall terms) what error message is returned.

Also, when we write 'explain your answer' we want specific and detailed reference to the Prolog execution model; you must convince the examiner that you understand how the Prolog abstract machine works.

(a) Consider the following two Prolog databases:

- Database 1:

```
cool(Prolog)
cool(Python)
cool(ML)
```

- Database 2:

```
cool(ice)
cool(snow)
cool(space)
```

State the output of the query `cool(X).` in the respective databases, and explain your answer. (5)

(b) Consider the following Prolog database:

```
greater_than(X,Y) :- greater_than(X,Z), greater_than(Z,Y).
greater_than(3,2).
greater_than(2,1).
```

1. State precisely the output of the query `greater_than(3,1).` in this database, and explain your answer with specific and detailed reference to the Prolog execution model. (4)
2. How could the design of the database be improved? (1)

(c) Consider the following Prolog database:

```
p(a).
p(b).
```

State the output of the query `p(X),!.` and explain your answer. (2)

(d) Consider the following Prolog database:

```
list_processor([],R):-e(R).
list_processor([H|T],R):-
   list_processor(T,TR),
   f(H,TR,R).
e(0/0).
f(X,A/B,A1/B1):-A1 is A+X, B1 is B+1.
```

1. State the output of the query

   ```
   list_processor([3,4,5,6,7,8,9],Y).
   ```

   in this database, and explain your answer. (4)

2. State the output of the query

   ```
   list_processor(X,100).
   ```

   in this database, and explain your answer. (2)

(e) "Prolog is a declarative language." Discuss this claim, giving one specific reason for, and one specific reason against. (2)

END OF PAPER